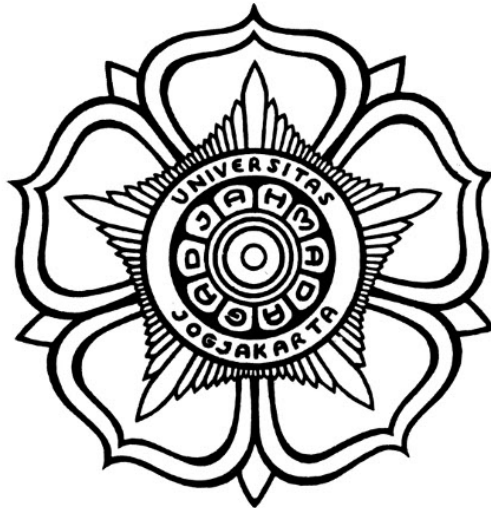


TUGAS

PERANCANGAN SISTEM DIGITAL



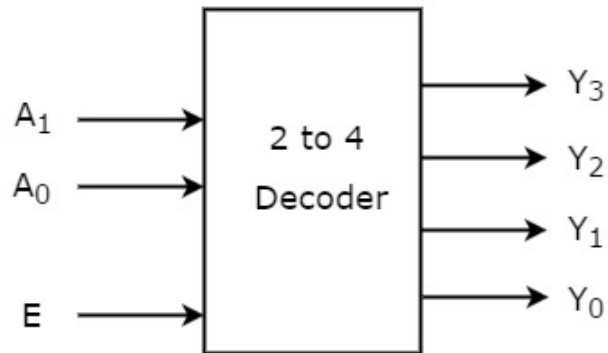
Disusun oleh:
AFRIZAL DANI SAOQI
17/413500/TK/45940

**DEPARTEMEN TEKNIK ELEKTRO DAN TEKNOLOGI
INFORMASI
FAKULTAS TEKNIK UNIVERSITAS GADJAH MADA
YOGYAKARTA**

2020

1 Decoder 2 to 4 using behavioral modeling

1. Architecture



Gambar 1: Architecture

2. Source Code

```

// Initialize verilog module
// stating input and output will be used
module decoder_24(
    input [1:0]w,
    input EN,
    output [3:0]out
);

// Stating behavioral model, output 'out' is stated by 'reg'
// The output is always preserved by the input 'w'
reg [3:0]out;
always @(w,EN)
begin
    if (EN==1'b1) //When Enable
        if (w==2'b00)
            out=4'b0001;
        else if (w==2'b01)
            out=4'b0010;
        else if (w==2'b10)
            out=4'b0100;
    end
end
  
```

```

        else if (w==2'b11)
            out=4'b1000;
        else
            out=4'bZZZZ; // for rare situation
        else // When disable , output=0
            out=4'b0000;
    end
endmodule

```

3. TestBench Source Code

```

// Simulating baseline time for simulation control
`timescale 1ns / 1ns
// Initialize module for test bench
module tb_decoder_24;
    reg [1:0]w; // Input circuit 2bit
    reg EN;

    // Output circuit
    wire [3:0]out; // ouput 4 bit

    // Instantiate
    decoder_24 decoder1 (w, EN, out);

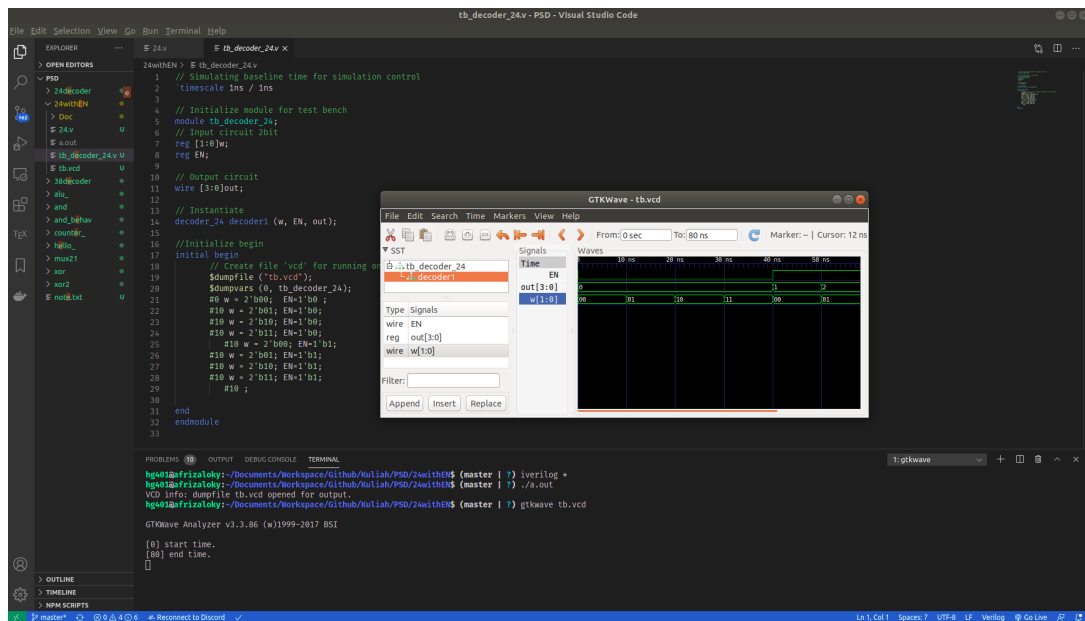
    //Initialize begin
    initial begin
        // Create file 'vcd' for running on gtkwave
        $dumpfile ("tb.vcd");
        $dumpvars (0, tb_decoder_24);
        #0 w = 2'b00; EN=1'b0 ;
        #10 w = 2'b01; EN=1'b0;
        #10 w = 2'b10; EN=1'b0;
        #10 w = 2'b11; EN=1'b0;
        #10 w = 2'b00; EN=1'b1;
        #10 w = 2'b01; EN=1'b1;
        #10 w = 2'b10; EN=1'b1;
    end
endmodule

```

```
#10 w = 2'b11; EN=1'b1;
#10 ;
```

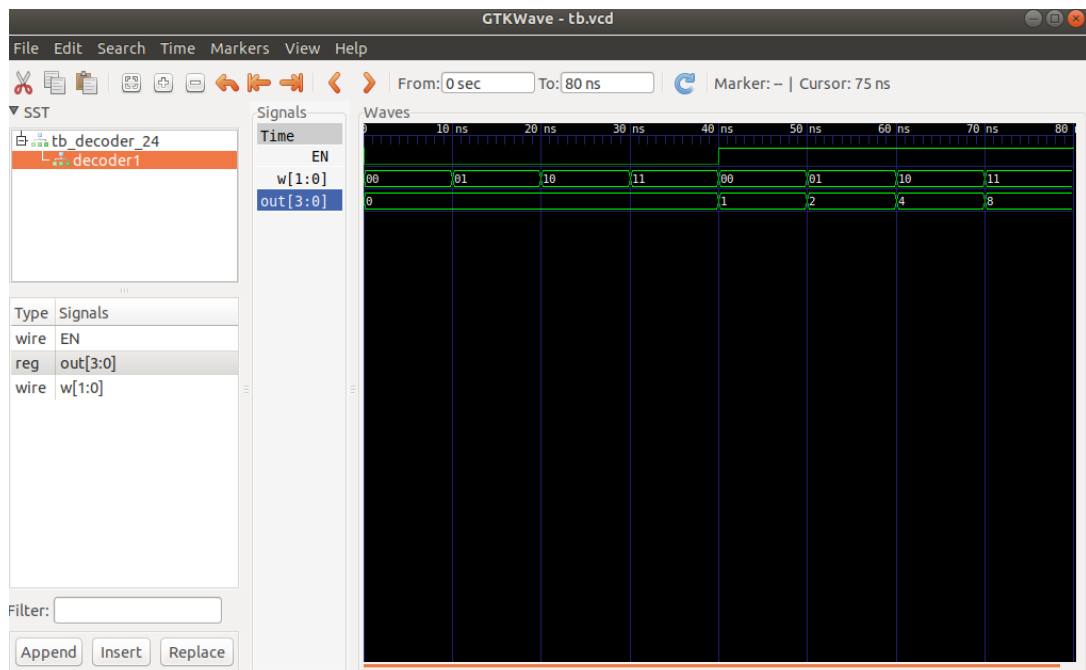
```
end
endmodule
```

4. Command Line



Gambar 2: Command Line

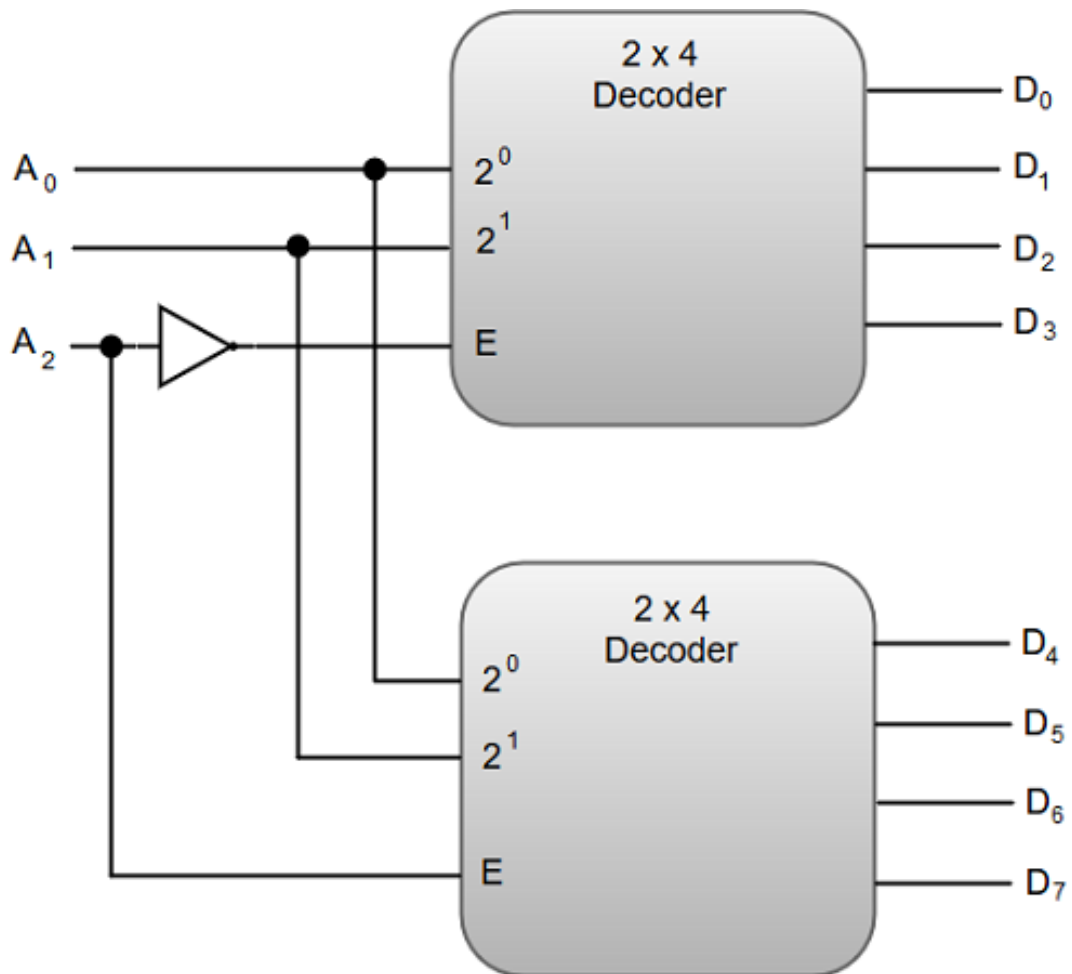
5. Simulation



Gambar 3: Simulasi Decoder 2 to 4

2 Decoder 3 to 8 using decoder 2 to 4

1. Architecture



Gambar 4: Architecture

2. Source Code 2 to 4

```
// Initialize verilog module
// stating input and output will be used
module decoder_24(
    input [1:0]w,
    input EN,
    output [3:0]out
);

// Stating behavioral model, output 'out' is stated by 'reg'
// The output is always preserved by the input 'w'
reg [3:0]out;
```

```

always @(w,EN)
begin
    if (EN==1'b1) //When Enable
        if (w==2'b00)
            out=4'b0001;
        else if (w==2'b01)
            out=4'b0010;
        else if (w==2'b10)
            out=4'b0100;
        else if (w==2'b11)
            out=4'b1000;
        else
            out=4'bZZZZ; // for rare situation
    else // When disable , output=0
        out=4'b0000;
end
endmodule

```

3. Source Code 3 to 8 using 2 to 4

```

module decoder_38 (
    input [2:0]w, // declaration 3 bit input
    output [7:0]out // declaration 8 bit output
);

    // instance module decoder 2 to 4
    decoder_24 dec0 ( w[1:0], ~w[2], out[3:0]);
    decoder_24 dec1 ( w[1:0], w[2] , out[7:4]);

endmodule

```

4. TestBench Source Code

```

// Simulating baseline time for simulation control
`timescale 1ns / 1ns

// Initialize module for test bench

```

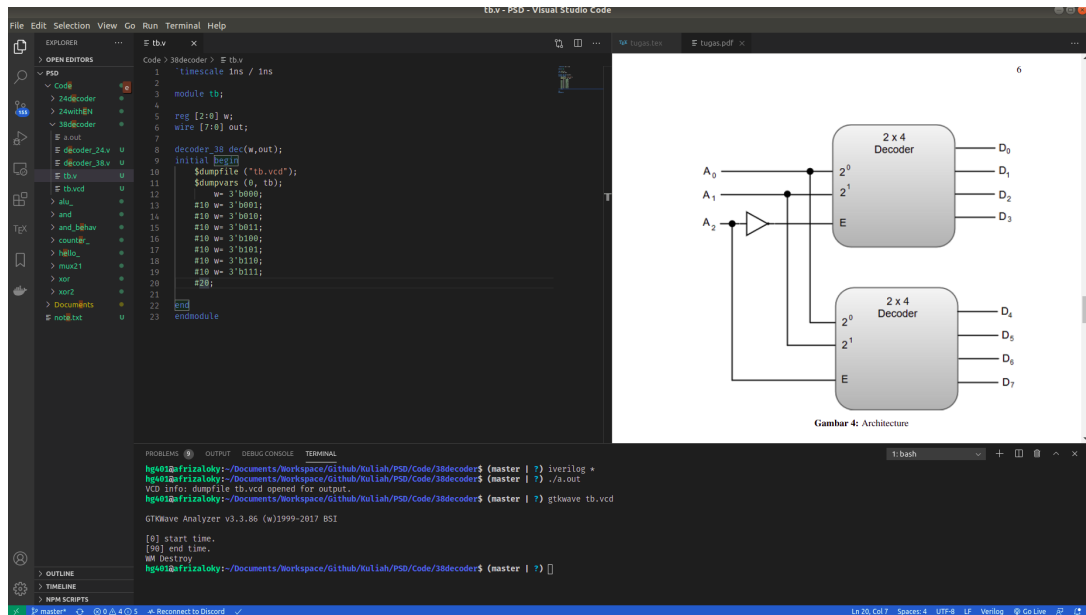
```
module tb;

reg [2:0] w; // input 3 bit
wire [7:0] out; // output 8 bit

decoder_38 dec(w,out); // instantiate
initial begin
    $dumpfile ("tb.vcd");
    $dumpvars (0, tb);
    w= 3'b000;
    #10 w= 3'b001;
    #10 w= 3'b010;
    #10 w= 3'b011;
    #10 w= 3'b100;
    #10 w= 3'b101;
    #10 w= 3'b110;
    #10 w= 3'b111;
    #20;

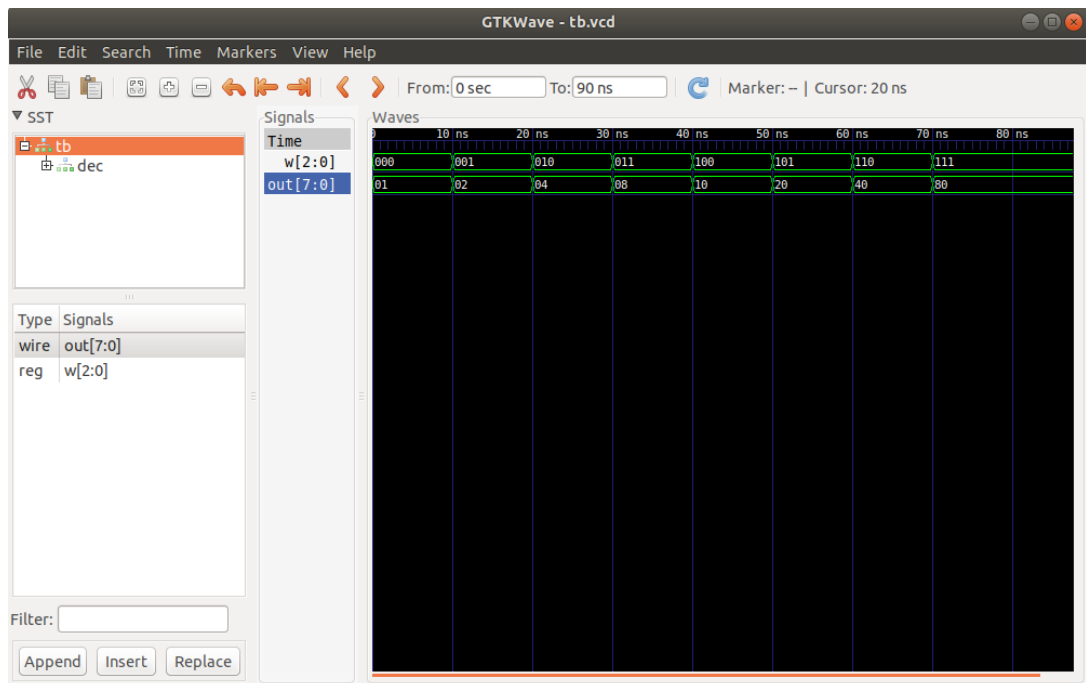
end
endmodule
```

5. Command Line



Gambar 5: Command Line

6. Simulation



Gambar 6: Simulasi Decoder 3 to 8

3 Repository

Github Repository